Digital Picture Processing

AZRIEL ROSENFELD

Computer Science Center
University of Maryland
College Park, Maryland

AVINASH C. KAK

School of Electrical Engineering
Purdue University
West Lafayette, Indiana

序　文

計算機による画像情報の処理についての最初の論文が出てから，もう約20年になる．それ以来今日まで，この分野は急速に発展してきており，英語による論文だけを数えても，(この分野で)年間500以上の論文が発表されていると推定される．これらの論文の多くは応用に関するものであるが，今日に至るまでにかなりの数の基本的な画像処理技術が開発されてきた．この本は，このような技術的観点から画像処理の分野を取り扱う．

この本は著者の1人(A. Rosenfeld)が前に出した本：電子計算機による画像処理(Academic Press, 1969年; 日本語訳，共立出版，1971年; ロシヤ語訳，Mir出版，1972年)にとってかわるものである．この本では画像のディジタル化，圧縮，および復元についてより徹底した取扱いをした．その部分(第4,5,7章)，および準備としての第2章はA. C. Kakが書いた．一方，この本では光学的(および他のアナログ的)処理方法については取り扱わなかった．また，3次元風景の解析，たとえば物体までの距離を測るための立体視技術や距離計技術，あるいは射影幾何の応用などに特有の問題も扱わなかった．さらに，計算機で合成された画像の処理についても書いていない．これはコンピュータ・グラフィックスの主要問題である．

この本のある部分，とくに第4,5,7章にはある程度の数学的基礎が必要である．とくに基礎として知っていることが望ましい分野は，線形システム理論(変換技術を含む)と確率論(確率変数，確率過程)である．これらの分野に対する入門の部分をもうけ，本として比較的自立性を保つことに意を用いた．この本が電気工学と計算機科学の両分野の学生に使われるためには，そうすることが必要であると感じた．

この本は学部高学年か修士課程の画像処理の1ないし2学期分として適当であろう．この本は多くの内容を含んでいるので，その中からいくつかの主題を選ん

とする．このとき $\varphi\bar{s}$ は，

```
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
```

$\varphi\bar{s}$ の4つの平行移動したものは，

```
1 0 0 0 1   0 0 0 0 0   0 1 1 1 1   1 1 1 1 0
1 0 0 0 1   1 1 1 1 1   0 1 0 0 0   0 0 0 1 0
1 0 0 0 1,  1 0 0 0 1,  0 1 0 0 0,  0 0 0 1 0
1 1 1 1 1   1 0 0 0 1   0 1 0 0 0   0 0 0 1 0
0 0 0 0 0   1 0 0 0 1   0 1 1 1 1   1 1 1 1 0
```

その結果 OR $\varphi\bar{s}^{U}$ は，

```
1 1 1 1 1
1 1 1 1 1
1 1 0 1 1
1 1 1 1 1
1 1 1 1 1
```

これと φ_S との AND は

```
0 0 0 0 0
0 1 1 1 0
0 1 0 1 0
0 1 1 1 0
0 0 0 0 0
```

であり，これがまさしく $\varphi_{S'}$ である．

演習 12. S の点 (i,j) は，S 中に近傍を持たないとき孤立(isolated)しているという．与えられた S の孤立点を見つける逐次型および並列型のアルゴリズムを記述せよ．

S' だけが与えられたとき S を作成するのは，一般には不可能である．たとえば，S' が


```
1 1 1
1   1
1 1 1
```

のとき，S が次のいずれであるか区別できる方法はない．

```
1 1 1          1 1 1
1   1  または  1 1 1
1 1 1          1 1 1
```

しかしながら，S' の点のどの近傍が S 中にあるかがわかっていれば，S を“埋め込む(fill in)”のは容易になる(9.1.2節参照)．

次の節では，与えられた S の境界の“追跡(follow)”のしかたを述べる．これを用いると，境界のチェイン符号の作成が容易になる．またこの方法は，境界点がきわめて少ないときは，単純な境界発見の走査法よりもはるかに計算コストを少なく済ますことができる．

9.1.2 境界追跡

C を S の成分，D を $\bar{S}$ の成分とする．D の点に隣接する C の点の集合を C の D-境界と表現する．これには，C, D が4-あるいは8-成分のいずれであるか，また4-あるいは8-隣接性のいずれを使うかによって，いく通りかの定義が可能である．以下では，C を8-成分，D を4-成分とし，4-隣接性を用いることを仮定しよう．他の場合に関しては演習14を見よ．

簡単な例として，C が

```
1 1 1 1
1   1 1
1 1 1 1
```

で，D が C の背景のとき，C の D-境界は

```
1 1 1 1
1     1
1 1 1 1
```

である．一方，D が C の孔のとき，D-境界は

```
  1
1   1
  1
```

である．(8-隣接性を用いるなら，

```
1 1 1
1   1
1 1 1
```

となる．) 2つの異なった D-境界が共通の点を持ち得ることに注意せよ．たとえば C が

```
1 1 1
1   1
1 1 1
```

のとき2つの D-境界は全く同じとなる．

演習 13. C の与えられた点は，何通りの異なる D-境界上の点と成り得るか．

C の D-境界の点をすべて見つけるアルゴリズムを述べよう．初めに1対の隣接点，C 中の c と D 中の d が与えられているとする．まず，C が唯一の孤立点でないかをチェックする．C が唯一の点 c しか持たないなら，この点が C の D-境界であり，これで終りである．そうでないとき，このアルゴリズム——BF (border following) と呼ぼう——は次のように進む．

(1) c, d の値をそれぞれ 3, 2 に変更する．

(2) c の 8-近傍を(たとえば)時計回りにたどり，d で始まり 1, 3, 4 のいずれかで最初のもので終わる系列を，$e_1, \cdots, e_k$ と示す．

(a) ある $h<k$ に対して，$c=3$, $e_k=4$, $e_h=2$ なら，この3を4に，2を0に変更し，停止する．

(b) これ以外のとき，c の値を(1であったなら) 4 に変更する．新たに e_k を c, e_{k-1} を d として，ステップ(2)へ戻る．

BF が停止したとき，4が C の D-境界の点となる．

以下は BF の動作の例である．C を


```
  *
1 1
1 1
    1
```

とする．ここで，最初の c は下線つきで，最初の d の位置は星印で表されている．

表1に，BF のステップ(2)における一連の状態を示す．ここでは，現在の c には下線をつけ，e_1 は星印で，$e_2, \cdots, e_{k-1}$ は $u, v, w, \cdots$ で表し，e_k には $'$ をつけてある．最後の状態では，$c=3$, $e_3=2$, $e_5=4$ であるので，この3を4に，2を0に変更して停止する．

別の例として次のパターンを考えよう．

表 1

ステップ(2)への入力	パターン	ステップ(2)への入力	パターン
1	2* u 3 1′ 1 1 1	5	2 3 4 1′ 4 u * 4
2	2 * u 3 4 v 1 1′ w 1	6	2 w 3′ 4 v 4 4 u * 4
3	2 3 4 1 4 * 1′	7	u 2 w * 3 4′ 4 4 4
4	2 3 4 1 4′ * u z 4 v y x w		

$$\begin{matrix} & & 1 \\ & * & \underline{1} \\ & & 1 \end{matrix}$$

連続的な段階を表2に示す．BFは第3の状態では停止しない．$c=3$であっても2がeの中に現れないからである．第4の状態では，$e_6=2$でも$c\neq 3$であるから停止しない．しかし，第5の状態では，$c=3$，$e_1=2$であるから停止する．

表2

ステップ(2)への入力	パターン	ステップ(2)への入力	パターン
1	$u\ \ v\ \ 1'$ $2^*\ \ \underline{3}$ 1	4	4 $2\ \ 3'$ $x\ \ \underline{4}\ \ *$ $w\ \ v\ \ u$
2	$u\ \ v\ \ w$ $*\ \ \underline{4}\ \ x$ $2\ \ 3'\ \ z\ \ y$ 1	5	$u\ \ v\ \ 4$ $2^*\ \ \underline{3}$ 4
3	4 $2\ \ \underline{3}\ \ *$ $1'\ \ u$		

演習 14.　Cが4-連結，Dが8-連結であり，やはり4-隣接性を用いる場合に対する修正版アルゴリズムBFを定義せよ．（**ヒント**：ステップ(2)において，e_kを値1, 3, 4のいずれかを持つ最初の4-近傍とせよ．e_{k-1}がこれらの値のいずれをも持たないときは，e_kを新たなcに，e_{k-1}を新たなdとせよ．）他の連結性と隣接性を用いるとどうなるか．

演習 15.　CのD-境界の連続した2点から始め，同じ順に連続なこの2点を再び見つけたとき停止するやり方での修正版アルゴリズムBFを定義せよ．

演習 16.　Cを4-連結8成分（すなわち，Cは4-連結で，かつ$S-C$のどの点もCに8-隣接していない）とする．CとDの1対の4-隣接点をそれぞれb_1, b_2とし，b_kを次のように帰納的に定義する．b_{k-1}がS中にあれば左に曲がり，$\bar{S}$中にあれば右に曲がる．（もっと正確には，b_{i-1}からb_iへの方向をθ_iとして，S中にb_{k-1}があれば$\theta_k=\theta_{k-1}+90°$，$b_{k-1}$が$\bar{S}$中にあれば$\theta_k=\theta_{k-1}-90°$とする．）このとき系列$b_1, b_2, \cdots$中の1は，$C$中の$D$-境界の点をすべて含む．$C$が4-連結でないなら，これは成り立たないことを示せ．（**ヒント**：Cを${}_1{}^1$とせよ．）

演習 17.　(c_i, d_i)をそれぞれC, D中の1対の隣接点とする．ここでCは4-連結，Dは8-連結であり，4-隣接性を用いる．e_i, f_iは，c_i, d_iと2×2の正方形を作る1対の点で，c_iを向かって左にとるとき手前側にあり，e_iはc_iと隣接，f_iはd_iと隣接しているものとする．$e_i=0$のとき$c_{i+1}=c_i$，$d_{i+1}=e_i$；$e_i=1$かつ$f_i=0$のとき$c_{i+1}=e_i$，$d_{i+1}=f_i$；$e_i=f_i=1$のとき$c_{i+1}=f_i$，$d_{i+1}=d_i$とせよ．この手続きは，Cの“壁を左側に見ながら”，CのD-境界に沿って，CとDの“裂け目(crack)”をたどる．別の種類の連結性に対して同様な手続きを定義せよ．

BFアルゴリズムを用いて，与えられた図形サブセットSのすべての境界点を見つける検出・追跡法を定義できる．

(1)　0のすぐ右か，列の左端に1か2が見つかるまで，図形をTVラスタ式に走査する．この点(i,j)は，まだ追跡されてない境界(D-境界としよう)上のSの境界点のはずである．

(2)　BFを用いてD-境界を追跡し，次のようにマークをつける．

(a)　左側の近傍としてDの点を持つとき，1または2を3に変える．

(b)　左側の近傍としてDの点を持たないとき，1を2に変える．

(c)　3は変化させずにおく．

(3)　BFが終ったとき，(i,j)に戻っていて走査を続けることができる．この走査が終わったとき，Sのすべての境界点は2か3のマークがついている．

2つの異なったマーク（2と3）を使わないのなら，ステップ(1)で左側の近傍だけでなく1のすべての近傍をチェックしなければならない．そうしないと，境界点をいくつか見逃してしまう．たとえば，Sを

$$\begin{matrix} 1 & 1 & 1 \\ 1 & \underline{1} & 1 \\ 1 & & 1 \\ 1 & 1 & 1 \end{matrix}$$

とすると，$\bar{S}$は背景領域Bと1点からなる孔Hで構成される．まずSを左上の角からとりかかる．B-境界をたどって3マークをつけるなら，下線を引いた1だけはマークをつけない（これはB-境界上にはない唯一のH-境界の点である）．そして，この1は左側に0を持たないから，決して発見されない．一方，2と3の両方を使うと，B-境界追跡の後，